Malware Analysis Practical 01 Report

# Part 1:

## When was each file was compiled?

Hint: use PEView.

Lab01-01.exe = 2010/12/19 Sun 16:16:19 UTC

Lab01-01.dll = 2010/12/19 Sun 16:16:38 UTC

As you can see each file was created within a minute of each other which suggests that they were created by the same author. Also DLLs cannot run independently so it is likely that the executable uses the DLL.

## Are each of the files packed or obfuscated?

Hint: use PEiD.

No Microsoft Visual Studio was used to compile both files.

The two files both have a small number of imports suggesting they are small programs. However, the DLL has no exports which is unusual but not indicative of the file being packed.

To conclude there is no evidence to suggest either file is packed.

## Examine the import functions of each file, is there any indicative of the functionality of the files and why?

Hint: use Dependency Walker

Lab01-01.exe uses some interesting imports that indicate to me that the program searches and copies files from the filesystem. The interested files/functions are shown below;

* FindFirstFileA
* FindNextFileA
* CopyFileA

Lab01-01.dll appears to create an object, runs a process, runs the created object, and then sleeps. This is interesting as it will sleep in the background until its sleep interval has passed in which it will run. It perhaps is hiding until a certain condition is met in order to execute. This DLL also has network functionality using imported WS2\_32.dll. We cannot see the functions being utilised by this DLL as they are ordinally imported. The interested files/functions are shown below;

* createProcessA
* sleep
* WS2\_32.dll

## Are there any host-based indicators?

Using strings.exe we can see there is a file similar to ‘kernel32.dll’ however the ‘L’ is changed to a ‘1’; ‘kerne132.dll’. This could be used to confirm is a machine has been infected.

## Are there any network-based indicators?

Yes, there is an IP address 127.26.152.13 used in the DLL file shown using strings.exe. This could be a good network-based indicator to identify this program as malware.

# Part 2:

Examine Lab01-03.exe

## Are there any indications that this file was packed?

There are a couple indications that this file was packed. Firstly, PEiD showed it was packed using ‘dulek/xt’. Secondly the raw size showed ‘00000000’ with the virtual size showing ‘00003000’ that is considerably bigger indicating it is compressed.

### What are the file imports?

The file imports kernel32.dll using functions ‘LoadLibraryA’ and ‘GetProcAddress’.

# Part 3:

Examine Lab01-04.exe

## Are there any indicators the file is packed?

There are no indications the file is packed. PEiD showed it was compiled using Microsoft Visual Studio C++ 6.0. The raw size was also bigger than the virtual size indicating that it is not compressed.

## When was the file compiled and what does this indicate?

The file was compiled in August 2019 at 22:26:59 UTC. We can’t take this creation time for certain as date stamps can be spoofed.

## What do the file imports tell us about its functionality?

This file imports ‘Kernal32.dll. that accesses memory, files and hardware along with ‘Advapi32.dll’ that accesses the registry or service manager.

The functions used within ‘Kernal32.dll’ tells us that the program is writing a file to the disk and executing it using functions ‘createFileA’, ‘WriteFileA’ and ‘WinExec’. It also may be moving directory using ‘MoveFileA’. There are also function imports for reading information from a resource in the file using ‘FindResourceA’, ‘LoadResource’ and ‘SizeOfResouce’. This resource could possibly be its malicious payload that it delivers once the program is executed.

The functions used within ‘Advapi.dll’ tells us that the program is doing something with permissions. For example, ‘adjustTokenPrivelageValueA’ enables or disables current set privileges. ‘LookupPrivilageValueA’ finds out what the current privilege of a user/file is. ‘OpenProcessToken’ contains security information of the current login session. This also tells us that the program is trying to access protected files most likely in the windows directory as ‘GetWindowsDirectoryA’ is used under the ‘Kernal32.dll’ import.

## Are there any host-based indicators that could identify this malware?

In strings there are two programs listed. The first is named ‘wupdmgrd.exe’ looks similar to ‘wupdmgr.exe’, the windows update manager program. The second listed is called ‘winup.exe’ that is most likely made to look like the windows update executable. These both could be used to identify an infected machine.

## Are there any indicators of network activity?

The URL ‘www.practicalmalwareanalysis.com/update.exe’ is present in one of the programs strings. This is most likely where the additional malicious payload is stored and downloaded from.